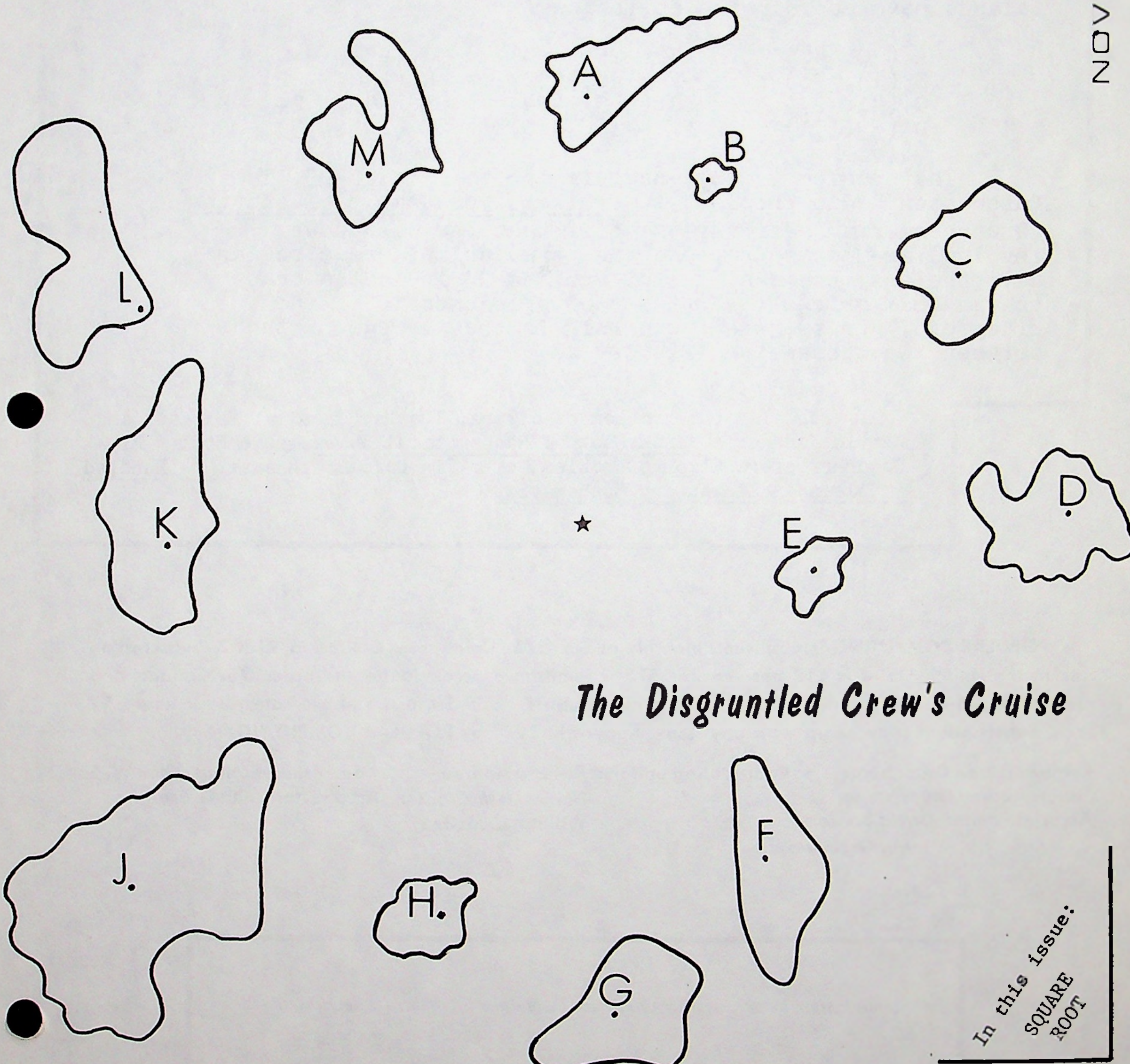
Popular Computing

20

VOL 2 NO 11

NOVEMBER 74



The Disgruntled Crew's Cruise

In this issue:
SQUARE ROOT

PROBLEM 68

The twelve islands of the Choofootse group have to be visited each day for mail and supplies by an air crew. The islands are normally toured in the order ABCD...LM, since that ordering provides the shortest trip.

During a labor dispute, the crew finds that the ordering of their daily trip is not specified, so they decide to visit the twelve islands in the order that will make the longest possible trip. What is that order?

From the point indicated by the star, the twelve islands have coordinates as follows:

A	(0, 48)	E	(25, -5)	J	(-50, -40)
B	(13, 38)	F	(20, -37)	K	(-46, -2)
C	(41, 27)	G	(3, -54)	L	(-49, 24)
D	(53, 1)	H	(-16, -43)	M	(-24, 39)

The problem is somewhat similar to Problem D3, The Pushbutton Radio Problem, in Problems for Computer Solution (Gruenberger and Jaffray), except that the distances involved are not equal, and the permutations are circular. Nevertheless, the analysis of Problem D3 indicates that the maximum trip between the twelve islands is of the order of 77 units, where one unit is the average distance between two successive islands.

See also the problem of O'Gara, The Mathematical Mailman, in Martin Gardner's Sixth Book of Mathematical Games, page 235. Mr. Gardner refers also to Problem No. 64 in Hugo Steinhaus' One Hundred Problems in Elementary Mathematics.

POPULAR COMPUTING is published monthly at Box 272, Calabasas, California 91302. Subscription rate in the United States is $18 per year, or $15 if remittance accompanies the order. For Canada and Mexico, add $4 to the above rates. For all other countries, add $6 to the above rates. Back issues $2 each. Subscriptions may begin with any issue. Copyright 1974 by POPULAR COMPUTING.

Publisher: Fred Gruenberger
Editor: Audrey Gruenberger
Associate editor: David Babcock
Irwin Greenwald
Contributing editors: Richard Andree
Daniel D. McCracken
William C. McGee
Advertising manager: Ken W. Sims
Art director: John G. Scott

Book Review ... W C McGee

```
      INTEGER W(15),Z,OW
      DIMENSION O(20)
      COMMON O
      WRITE(6,91)
   91 FORMAT (30H1FORTRAN: A DISCOVERY APPROACH/
     1        17H BY ROBERT TEAGUE/
     1        20H CANFIELD PRESS 1974//
     1        22H REVIEWED BY W.C.MCGEE//)
      CALL MP(1,0,0,0,0)
      OW=0
      N=0
   10 READ (5,92)(W(I),I=1,15)
   92 FORMAT (15I5)
      DO 80 I=1,15
      Z=W(I)
      IF (Z) 20,100,30
   20 Z=-Z
      N=1
   30 NCH=MOD(Z,10)
      NOW=Z/10
      IF(NOW.EQ.0) GO TO 40
      CALL MP(2,1+MOD(NOW-1,12),1+(NOW-1)/12,NCH,OW+1)
      OW=OW+NCH
      GO TO 50
   40 CALL MP(3,NCH,0,0,OW+1)
      OW=OW+1
   50 IF(N.EQ.1) GO TO 60
      WRITE (6,93) (O(J),J=1,OW)
   93 FORMAT (1H ,20A1)
      OW=0
   60 N=0
   80 CONTINUE
      GO TO 10
  100 STOP
      END

      SUBROUTINE MP(SW,I,J,N,K)
      INTEGER SW,BJ,R,C
      DIMENSION TEXT(12,12),OUT(20),PUNC(4)
      COMMON OUT
      DATA PUNC /1H.,1H,,1H',1H-/
      BJ(I,J)=(I+(J-1)*12)/12+1
      GO TO (10,20,30), SW
   10 READ (5,11) TEXT
   11 FORMAT (12A1)
      RETURN
   20 R=I
      C=J
      DO 25 L=1,N
      M=K+L-1
      OUT(M)=TEXT(R,C)
      C=BJ(R,C)
      R=MOD(R,12)+1
   25 CONTINUE
      RETURN
   30 OUT(K)=PUNC(I)
      RETURN
      END
```

(Data for this Fortran program is on the next page.)

(Data for this Fortran program is on the next page.)

```
VENDISHICONS
ARESEDBOOKEA
RKPITIONIQUE
ACHAPTERCLEN
TOFASSPOSTSE
SOCOMPROGRAM
MINGODCETHER
ATIVEXWHINAU
TUDEMONMANGI
CICALANGUAGE
FORTRANDPLIU
SIBLLYTYWOBY
  813 -762 1122  562 -812   52  194   52  492 -491  364 -634 -962 -743    2
-1322  743  813 1217  872 -668  743 1138  522  813 -271 -172-1261 -762-1072
 1113  -12  -72 -453    1 1261 -592 -732   -4 -613 -843 1082 -422 -382  325
   52-1322  172  742 -913  382   -3 -121 -233 -382  743 -343 -572 -303-1331
    3  133 -553  172 1263-1392 -272 1113 -572 -982  473 -142 -542 -133  152
  133-1072  -13    1  813 1138 -892 -553  285   52  -93  -13 -294-1122    2
-1201   13  492  813  -43-1292 -851 1441  562  851 -633-1292 -802  301  667
 1432  813   23  502  387-1341    1-1321  325-1211 -232 -972  143  133  951
  -43 -451-1171 -532  363  502 -633 1013 -668  743 -273 -512-1362-1331    2
 1263 -923  582  152 -762  772 1217 -663 -698    1 -813  142  133-1343 1441
-1081-1642-1072-1061  855 -892 -832-1692  593 1263 -663-1352-1032 1331  742
  364 -387    1  131-1412 -252-1043-1362  222-1202-1214    2  813  194 -995
 -572 -843  152 -812  852  813 -952 -812-1222   -3 1331   -3 -692-1131 -542
    3 -452 -233 1372 -892 -792 -172 1331  813   -3 -701 -364    3  562-1441
  802-1152 -771  814 1217 -194    1
```

Mr. McGee's Fortran program outputs a review of Robert Teague's Fortran text. As one might suppose, the review is quite favorable.

Speaking of Languages

ROBERT TEAGUE

The column this month is short due to space restrictions. It would, however, be a shame to let the above book review go by without comment. The fact that the text and this column have the same author is not what is noteworthy, but rather the unusual form of the review.

The Fortran program was written for IBM equipment and uses some non-standard techniques to produce the end result. This use of Fortran would appear to be unique, which might justify the use of these features. The program consists mainly of input/output statements, which read data cards and produce a printed listing that comprises the review. Take a close look at how the alphabetic material is produced; there is something to be learned from this exercise.

All in all, this is not only a novel, but highly appropriate and educational means of producing a book review.

MORE THAN YOU EVER WANTED TO KNOW ABOUT square root

Square root, the most elementary of functions, can be calculated by many algorithms. (In what follows, the square root of 153 will be used as an example; its value is

12.369316876852981649464229567922...)

1. The binomial theorem method. This is the algorithm that has been taught in our schools for hundreds of years, causing millions of students to hate arithmetic, and fear square root. The process is shown in Figure A.

The number whose root is sought is divided into periods of two digits, both ways from the decimal point. For the leading period, a trial square root is taken (which requires knowing the squares up to 100). The trial root is squared and subtracted from the first period. The root up to this point is now doubled and used as a trial divisor. From here on, the process somewhat resembles long division. In the example, 20 is to be divided into 53, but with the restriction that the quotient is to be first added to the trial divisor. Thus, 53 divided by 20 is 2, but the trial divisor changes to 20 + 2, as shown. New periods are brought down as needed, and the process repeats, getting progressively more awkward.

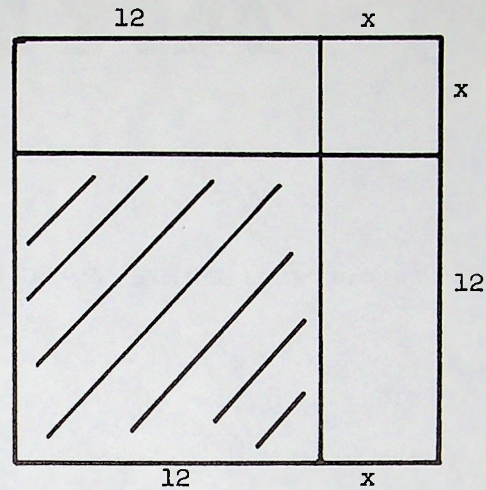
The geometric analog is shown, at the stage where the root has been located up to the decimal point, in Figure B. The large square has an area of 153 units, and the square of 12 (crosshatched) can be removed. What is left consists of two parts. We are expressing 153 as the square of something:

$$153 = (12 + x)^2 = 144 + 2 \cdot 12 \cdot x + x^2$$

A

```
           1  2 .  3  6  9  3
        √ 1`53 . 00`00`00`00
 20         1
  2        ____
 22        | 53
             44
 240        ____
   3       | 9    00
 243         7    29
  2460      _________
     6     |1     71 00
  2466      1     47 96
   24720       __________
       9       | 23 04 00
   24729         22 25 61
                _________
                | 78 39 00
```

B



and the x^2 term is disregarded as momentarily unimportant. The doubling in the algorithm is the 2 of the binomial expansion. We seek the width of the strips that represent $2 \cdot 12 \cdot x$. When we subtract 144 from 153, the difference of 9 is made into 900 by bringing down another period, and we "divide" by 24 to determine the width. As with many other methods, at all stages we really deal with the <u>sequence</u> of the digits involved; that is, we turn the problem into a problem in integers at all times.

The process is awkward and inefficient and requires memorizing an abnormal number of arbitrary rules. Since most seventh grade teachers have no idea why it works (but who recall that they too hate it), this fear and loathing is readily communicated to the students. And this is all quite mysterious, inasmuch as the algorithm given below has been around for hundreds of years and is superior in every way.

2. <u>Newton's Method</u>. This algorithm, which is the Newton-Raphson scheme for finding the root of an equation (applied to $x^2 - N = 0$) is usually given as an iterative formula:

$$x_{i+1} = (1/2)\left(\frac{N}{x_i} + x_i\right)$$

The method predates Newton by many years. It can be explained in these terms:

$$p \overline{\smash{)}N}^{\;q}$$

Using division, the number N is divided by any trial divisor, p, yielding a quotient q. If p = q, then, by the definition of square root, p is the desired root. In general, p will not equal q, and therefore one of p or q is less than the square root and the other is greater; that is, the root has been located between p and q. Any value between p and q will be a better approximation to the root and, lacking any other information, we take (p+q)/2 as the next approximation (what Newton showed, in fact, was that (p+q)/2 is the best choice that can be made simply.) The process now repeats, using (p+q)/2 as the new trial divisor. The arithmetic for N = 153 is shown in Figure C, using 12 as a starting value.

```
      1275
12 | 153
     144
      90
      84
       60
```

$$\frac{1200 + 1275}{2} = 1238$$

```
          123586
1238 | 153000000
       1238
        2920
        2476
         4440
         3714
          7260
          6190
          10700
           9904
            7960
            7428
             5320
```

$$\frac{123800 + 123586}{2} = 123693$$

C

Of the two possible sequences for the square root of the given sequence, we have established the one we want; hence, the decimal points are of little consequence in the calculation.

The Newton algorithm has the following advantages:

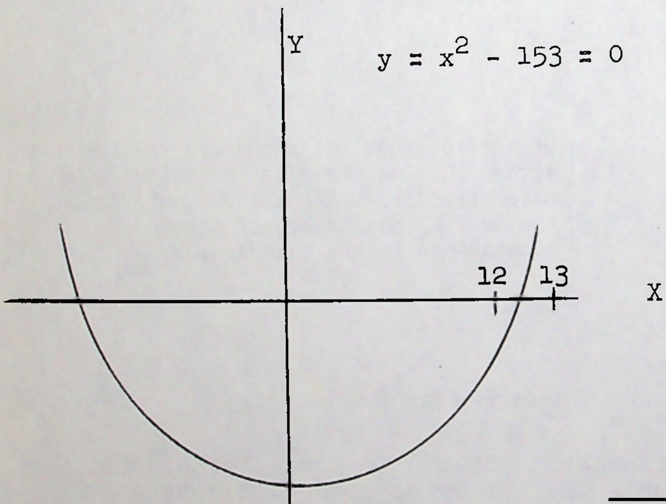
a. The process is fast. Stated in crude terms, the method allows us to double the number of correct digits at each stage. For each division, the quotient should be calculated to twice the number of digits that repeat. For example, in Figure C, the digits 12 repeat in the first division, and hence the quotient is carried to 4 digits. The result is not quite good to 4 digits, as seen in the second stage, where the digits 123 repeat. The second quotient is carried to 6 digits and, after interpolation, the result should be good to about 6 digits, which it is.

b. Since the procedure is iterative, it is also self-correcting. If an error is made, then the result at that stage is simply not as good as it might be.

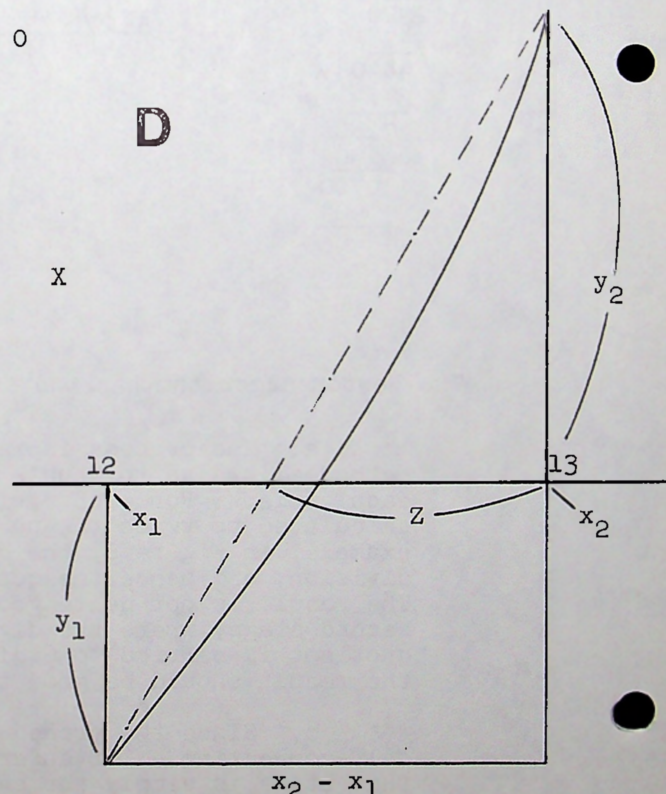
c. The procedure stems directly from the definition of square root, and hence has few rules to remember.

d. The scheme capitalizes on previous information. For example, if it is known that the square root of 2 begins with 1414, then that value can be used as the starting value, and one division (and interpolation) will yield 8 digits of the root. Thus, using a mechanical desk calculator and a one-page table of roots good to 4 significant digits, any square root can be obtained to the machine's limit (10 digits) with two divisions. And since the algorithm is based on division, it utilizes the part of the desk calculator for which the most money was spent.

e. The method is readily programmed for any computer, in any language. To avoid the problems of judgement involved in selecting a suitable starting value and in testing for convergence of the process, it is expedient to use either <u>one</u> or N as the starting value (which really makes the initial value (1+N)/2 and taking a fixed number of iterations, with each division carried out to the precision required of the result. Thus, in Fortran, using 8-digit floating arithmetic, six iterations will handle the worst case, using the starting value above.



D



E

3. <u>Other approximation methods.</u> Any of the well-known schemes for locating a root of an equation can be applied to the equation $x^2 - N = 0$. Horner's method is out of style, but the false position method is still used. See Figures D and E. We attempt to locate the point at which the curve $y = x^2 - 153$ crosses the x-axis. The region between $x = 12$ and $x = 13$ has been greatly enlarged. On the assumption that the straight (dotted) line is an approximation to the curve, the similar triangles involved imply that

$$\frac{Z}{x_2 - x_1} = \frac{y_2}{y_2 + y_1}$$

where all the values are taken as positive. We have then

$$x_2 - x_1 = 1, \quad y_2 = 13^2 - 153 = 16, \quad y_1 = 153 - 12^2 = 9.$$

From the relation above, $Z = 16/25$, or .64, and our next approximation is then 12.36. It is not at all clear to what extent we have located the root, but we can calculate a few values of the function, and determine that the function changes sign between 12.36 and 12.37. The process can now repeat, using those values as x_1 and x_2. The false position method seems best suited to hand calculation.

4. <u>Logarithms</u>. If we divide the logarithm of 153 by 2:

$$\log 153 = 2.18469143081759880313022\ldots$$
$$\log \sqrt{153} = 1.09234571540879940156011\ldots$$

the antilog of the result, found by interpolating in a 7-place table of logarithms (Vega's table) is 12.3693.

5. <u>Hastings Approximations</u>. Both the logarithm and the antilogarithm can be obtained from the ingenious rational functions found in Cecil Hastings' book, <u>Approximations for Digital Computers</u> (Princeton University Press, 1955). The logarithm given in (4) above was determined by direct (series) calculation of the logarithm. Hastings gives (his Sheet 20) the following approximation:

$$\text{for } 0 \leq x \leq 1$$

$$10^x \sim (1 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_7 x^7)^2$$

where
$a_1 = 1.15129277603$
$a_2 = .66273088429$
$a_3 = .25439357484$
$a_4 = .07295173666$
$a_5 = .01742111988$
$a_6 = .00255491796$
$a_7 = .00093264267$

We have here x = .09234571541. By substituting into Hastings' formula, the square root of 153 is found to be 12.3693168.

6. Table lookup. This is the first algorithm that most people meet; namely, to find a table of square roots and look up the desired answer. If a table of roots is stored in a computer, the table can be searched (linearly, or by binary searching), or the address of the required function value can be itself calculated. For example, if a table of 1000 entries (000 through 999 for the argument) is stored in a block addressed at T, then the functional value for 153 is at address T+153.

7. Interpolation. Given a table of square roots and differences, as follows:

N	SQRT	Successive differences		
153	123693169			
154	124096736	(+) 403567		
155	124498996	402260	(-) 1307	
156	124899960	400964	1296	(-) 11
157	125299641	399681	1283	13
158	125698051	398410	1271	12

we can find the square root of other values within the same range by utilizing well-known schemes for interpolation. To find the square root of 153.5, for example, Newton's forward interpolation formula specifies that

$$\frac{x - x_0}{h} = u = \frac{153.5 - 153}{1} = 1/2$$

$$f(u) = \sqrt{153.5} = y_0 + u\Delta y_0 + \frac{u(u-1)}{2}\Delta^2 y_0 + \frac{u(u-1)(u-2)}{6}\Delta^3 y_0 + \cdots$$

where the Δ's indicate successive differences. We have here

$$f(u) = 123693169 + (1/2)(403567) + (-1/8)(-1307) + \cdots$$

$$= 123895114$$

which is correct to 9 significant digits.

8. Bracketing methods. In a computer, the range of numbers to be considered is always restricted. For example, if we are working in fixed point on a machine with a 24-bit word, then the possible range on N is from zero to 8388607, which means that any square root must lie between zero and 2896. A perfectly straightforward algorithm, then, would be a loop to test the square of every value between zero and 2896 to determine which one most closely yields N. This would be terribly inefficient. The procedure shown in Figure F will speed up the process by a crude form of interval-halving. The resulting algorithm is discovered independently by countless sixth graders who want square roots but have not yet been exposed to any algorithm, even the miserable one give in (1) above.

As shown on the flowchart, the range on N has been further restricted to 20 bits, and the calculation for N = 15300 produces these numbers:

LØ	TRY	HI
0	0	999
0	499	999
0	499	499
0	249	249
0	124	124
0	62	124
62	62	124
93	93	124
108	108	124
116	116	124
120	120	124
123	123	124

and when the process converges, the root has been located within the limits of the specified precision. The algorithm is quite inefficient, but has two advantages:

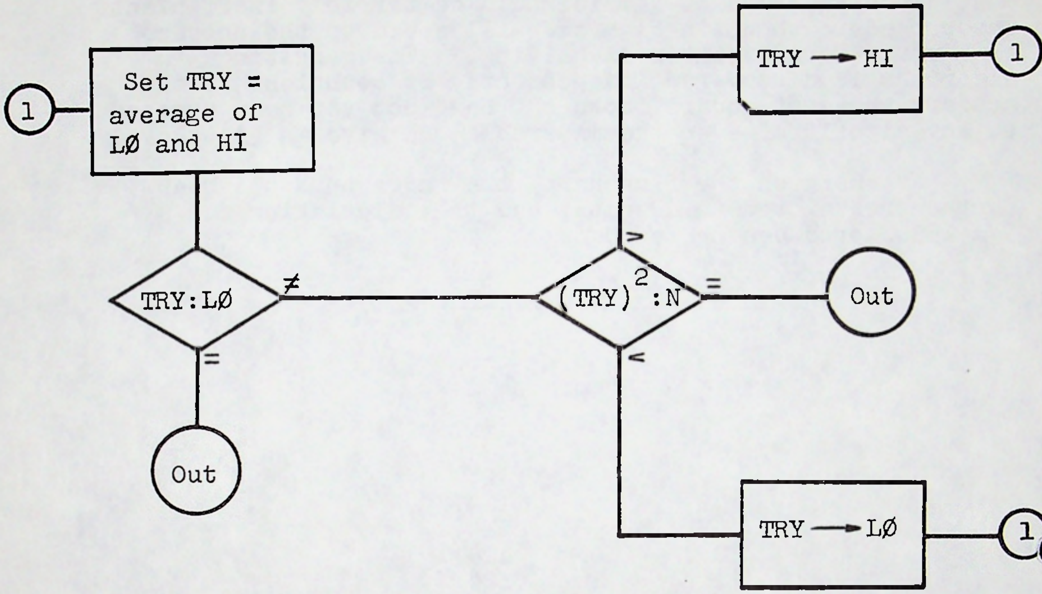
a. It is quite easy to program.

b. It uses no division. The "averaging" step can be done by multiplying by .5, or, in a binary computer, by a right shift of one bit position.

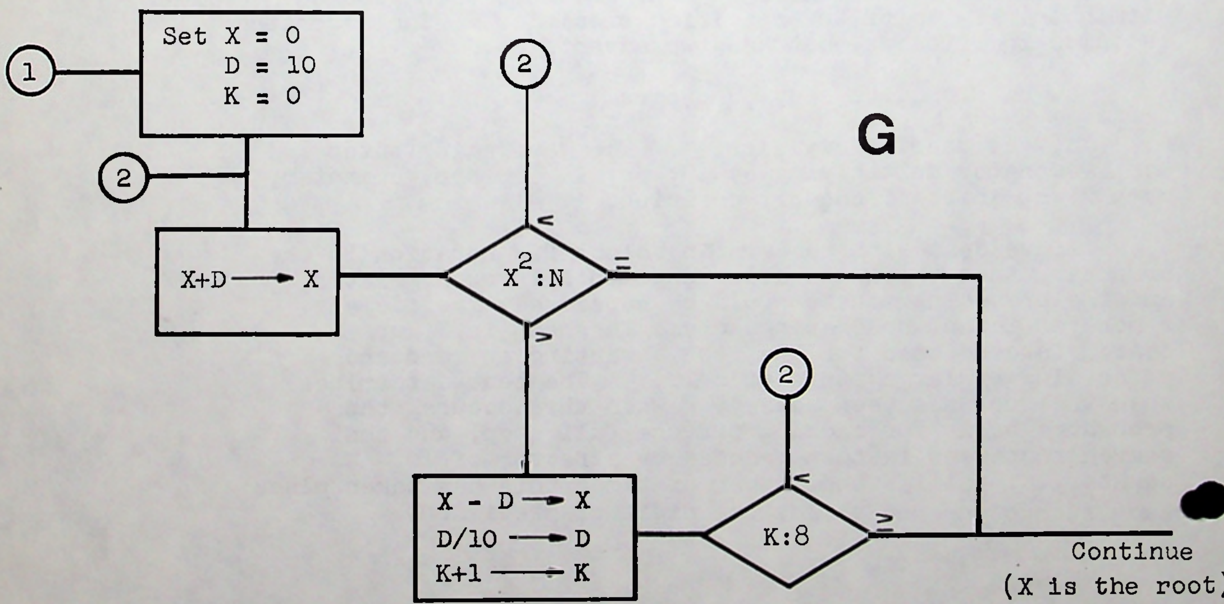
If we deal with numbers in scientific notation in the computer, then the admissable range on N is much greater, and the preceding scheme would be hopelessly inefficient. A more efficient bracketing scheme is shown in Figure G. A search is conducted for the root, starting at zero and proceeding by large steps (D = 10). The test determines when the root has been passed. When this occurs, the procedure calls for backing off one full step, and the search increment is then reduced by a factor of 10. A counter, K, tallies the number of times this has taken place; each such occurence yields one digit of precision.

Housekeeping: Set LØ = 000
TRY = 000
HI = 999

F



G



The algorithm of Figure G is also easy to program, and also uses no divisions. It is shown on the flowchart in general form. In practice, the value of the search interval and the initial search point should be functions of N, to cut down the machine time.

9. Odd number subtraction. From a table of differences of consecutive squares, or from the identity:

$$1 + 3 + 5 + 7 + 9 + \cdots + (2k - 1) = k^2$$

it can be seen that the sum of consecutive odd numbers equals the square of the consecutive integers. This fact can be used in reverse. By subtracting the odd numbers from a number whose square root is desired, and counting the number of subtractions, the root can be obtained. For example, for N = 153, successive subtractions of odd integers produce this pattern:

start	153	count the subtractions:
-1	152	1
-3	149	2
-5	144	3
-7	137	4
-9	128	5
-11	117	6
-13	104	7
-15	89	8
-17	72	9
-19	53	10
-21	32	11
-23	9	12
-25	underflow	

indicating that, to the nearest integer, the square root is 12. To extend the process, one begins with 15300 or 1530000. The process works from left to right, similarly to long division.

Before the days of automatic division on desk calculators, the odd number subtraction method was widely used for square root, particularly on key-driven machines (e.g., the Comptometer). Around 1952, the Friden company tried to mechanize the process on the first square root desk calculator, but the result was a failure. The user of that machine would enter a number into the lower dial, using the dividend entry key. When the square root key was depressed, the machine would start subtracting successive odd numbers from the lower dial. When that dial rolled negative, the machine would reverse and add one cycle (just as in division), shift the carriage, and begin subtracting odd numbers again. The trouble with the procedure was that two banks of the keyboard mechanism had to be controlled, as in the example we have seen. The mechanical details of this procedure were complex, and the machine broke down rapidly and frequently.

The solution to the problem, which appeared around 1954 in a new Friden square rooter, was ingenious. When the square root key on the new machine was depressed, the machine first multiplied N by 5, and then proceeded to subtract odd multiples of 5. The process is mathematically the same, but mechanically simpler, since the series of numbers 05, 15, 25, 35, 45,...,95 can be generated by locking a 5 in one bank of the keyboard and then generating consecutive integers in the bank to its left. The later machine worked fairly well, and for years was the only desk machine on the market with square root capability.

10. The quadratic method. The roots of the quadratic equation

$$x^2 - x + M = 0 \qquad \text{(A)}$$

can be found by the quadratic formula. If the roots can also be found by some other means, then, since the formula involves a square root, there results a very roundabout way of taking square roots. The equation (A) can be transformed into an interative scheme:

$$x = x^2 + M$$

$$x_{i+1} = x_i^2 + M \qquad \text{(B)}$$

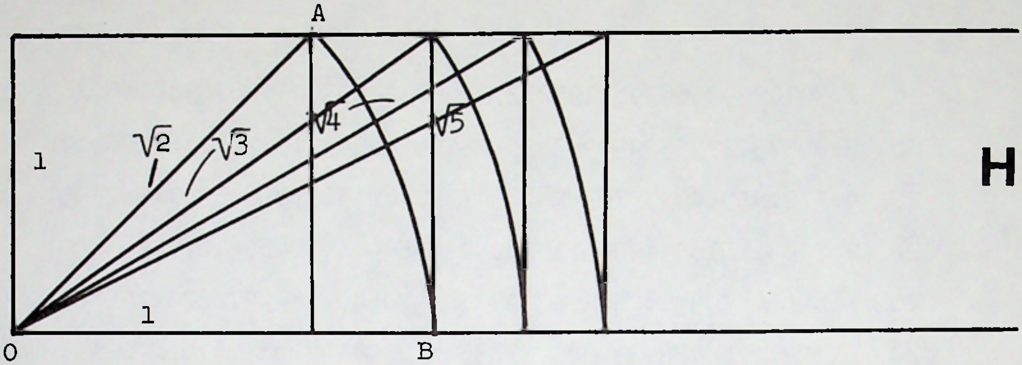
To find the square root of a number N, form the quantity

$$M = \frac{1 - N}{4}$$

with N chosen in the range from zero to one (which is not a serious restriction, since we seek the sequence of digits for the root and can adjust the decimal point later).

Thus, to find the square root of 153, take N = .0153. Then M = .246175. That value for M is used in equation (B), with some suitable starting value for x, and the process is iterated to convergence. When it converges, the required root will be (1 - 2x). The convergence rate is abysmally slow. The process seems useless as a practical method for finding square roots; it is included simply because it is a method.

11. Graphical methods. Figure H shows how any square root can be built up geometrically. The diagonal of a unit square is the square root of 2. If that length is brought down (arc AB, centered at O) and a perpendicular is erected, then the new diagonal is the square root of 3, and so on. Again, this method is somewhat impractical, but is another algorithm for square rooting.



12. The binomial theorem again. For numbers that differ from a perfect square by a square, the binomial theorem gives another method for square root. For example,

$$\begin{aligned}(144 + 9)^{1/2} &= (144)^{1/2} + (1/2)(144)^{-1/2}(9) \\ &\quad + \frac{(1/2)(-1/2)}{2}(144)^{-3/2}(9)^2 \\ &\quad + \frac{(1/2)(-1/2)(-3/2)}{2\cdot 3}(144)^{-5/2}(9)^3 + \cdots \\ &= 12 + .375 - .0058593 + .0001831 - .0000071 + \cdots \\ &= 12.3693167\end{aligned}$$

13. Some of these schemes can be combined. For example, table lookup can be used to furnish a good starting value for any of the iterative methods. For many years, the makers of mechanical desk calculators furnished one-page tables of square roots, with directions for extending the precision by the Newton method.

Every subroutine library and every programming package has a square root method built in. Which method is used is partly a matter of taste on the part of the author of the package. In a floating point system, it is most probably a Hastings approximation, perhaps followed by one stage of the Newton method.

Users of computers are indebted to many brilliant men who have brought the art of computing to its present level. In particular, though, we all use the work of three men constantly: John von Neumann, who started it all; Jay Forrester, who invented the magnetic core; and Cecil Hastings, Jr., whose approximations are at work in every function subroutine package.

PC20-16

Log 20	1.3010299956639811952137388947244930267681898814621O9
Ln 20	2.9957322735539909934352235761425407756766016229890 28
$\sqrt{20}$	4.4721359549995793928183473374625524708812367192230 51
$\sqrt[3]{20}$	2.7144176165949065715180894696794892048051077694890 97
$\sqrt[5]{20}$	1.8205642030260802643794210547054629849376874279588 45
$\sqrt[7]{20}$	1.5341274046343909812778351272954148281534165072290 19
$\sqrt[10]{20}$	1.3492828476735633151222197058090327666918884491375 95
$\sqrt[100]{20}$	1.0304105579112524899709294729483564750096257272046 51
e^{20}	485165195.40979027796910683054154055868463898894484725435361080031597799614270974016597985065275
π^{20}	8769956796.0826994747522555937038970660641144471954372434209842605184123904354799099023498518 67
$\tan^{-1} 20$	1.5208379310729538578213154046049065606073076192640 46

N-SERIES 20